UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,743	09/30/2003	Anandhi Somasekaran	MSFT-2763/305222.1	7938

41505 7590 06/25/2009

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
WANG, BEN C	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
06/25/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/676,743	SOMASEKARAN ET AL.	
	Examiner	Art Unit	
	BEN C. WANG	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *16 April 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20,23-35 and 37-65* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20, 23-35, and 37-65* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's amendment dated April 16, 2009, responding to the Office action mailed January 16, 2009 provided in the rejection of claims 1-20, 23-35, and 37-65, wherein claims 1, 7, and 12 have been amended.

Claims 1-20, 23-35, and 37-65, remain pending in the application and which have been fully considered by the examiner.

Applicant's argument with respect to *direct channel communication* (recited on 2nd – 3rd paragraphs, page 14, in the REMARKS) in the claims has been fully considered but is not persuasive. Please see the section of "Response to Arguments" for details.

Further, Applicant's other arguments with respect to claims currently amended have been fully considered but are moot in view of the new grounds of rejection – see *Casati et al. and Moser et al.* - arts made of record, as applied hereto.

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-11, 40-45, and 60-65 are rejected under 35 U.S.C. 103(a) as being unpatentable over O'Farrel et al. (Pub. No. US 2004/0168155 A1)

(hereinafter 'O'Farrel') in view of Casati et al. (Pub. No. US 2003/0236689 A1) (hereinafter 'Casati' - art made of record)

3. **As to claim 1** (Currently Amended), O'Farrel discloses a business process service debugger for remotely debugging a business process service (e.g., [0010] - ... a debugger allows debugging of applications using flow paradigm ...; [0011] - ... debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code ...), comprising:

- means for establishing a communications connection with a remote computer (e.g., Fig. 9; [0083] - ... flow debugger 64 ... to attach to a known one of computing devices ... executing run-time code ... The computing device may be local to flow debugger 64 or remote, identified by a network address ...);
- means for displaying a symbolic representation of the operation of the business process service based on the stored state information (e.g., [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17); and
- means for remotely debugging the one business process service using the symbolic representation, communications connection and stored state information (e.g., Fig. 9; [0083] - ... flow debugger 64 ... to attach to a known one of computing devices ... executing run-time code ...

The computing device may be local to flow debugger 64 or remote, identified by a network address ...)

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Analyzing Decision Points in Business Processes*, Casati discloses:

- means for indirectly communicating with a remote computer implementing the business process service, permitting the debugger to remotely debug the business process service without prior knowledge what remote computer it is running on (e.g., Fig. 5, elements 60 – the component of business platform; 70 – Web Server; 72 – Application Server (i.e., indirectly communicating with a remote computer implementing the business process service); 90 – Business Operational Intelligence Engine; 100 – Business Process Data Warehouse; [0046], Lines 8-19 - ... Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 ...);

- means for reading stored state information regarding historical events related to at least one business process implemented for the business process service (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 (i.e., the debugger) enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...);

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Casati into the O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

The motivation is that it would further enhance the O'Farrel system by taking, advancing and/or incorporating Casati's system which offers significant advantages for a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision points in the business process as once suggested by Casati (e.g., [0010])

4. **As to claim 2** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger, wherein business

processes and instances of the business process service other than those being debugged are not disrupted during debugging (e.g., [0011] - ... debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code ...)

5. **As to claim 3** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein the symbolic representation comprises a workflow of at least one business process in the business process service (e.g., Figs. 12, 13A, 13B, 14)

6. **As to claim 4** (Previously Presented) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger further comprising means for interacting with the business process service according to a user instruction (e.g., [0012] - ... receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)

7. **As to claim 5** (Original) (incorporating the rejection in claim 1), Casati discloses the business process service debugger wherein the stored state information corresponds to a variable assignment within the business process service (e.g., [0056] - ... for a certain set of values for process variables ...; [0062] – [0065] - Data variables)

8. **As to claim 6** (Previously Presented) (incorporating the rejection in claim 1), Casati discloses the business process service debugger wherein the events are historical events that occurred prior to failure of the at least one business process (e.g., [0046], Lines 8-19 - … Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 …)

9. **As to claim 7** (Currently Amended) (incorporating the rejection in claim 1), Casati discloses the business process service debugger wherein the stored state information corresponds to message flow data and an order in which run time components performed the at least one business process as a result of message processing (e.g., [0046], Lines 8-19 - … Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 …)

10. **As to claim 8** (Original) (incorporating the rejection in claim 1), Casati discloses the business process service debugger wherein said reading means further comprises means for reading stored business process service configuration information (e.g., [0029] - ... The appropriate resources may be selected by a resource executive based upon business logic that may be included as part of the process definition, work node definition, or system configuration ...; [0030] - ... Workflow management system 74 is configured to automate the execution of business processes ...)

11. **As to claim 9** (Previously Presented) (incorporating the rejection in claim 1), Casati discloses the business process service debugger wherein the events are events that occur prior to an inserted breakpoint in the one business process (e.g., [0046], Lines 8-19 - ... Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 ...)

12. **As to claim 10** (Original) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein said debugging means comprises means for detecting a location where the instance is being processed

(e.g., [0045] - ... in a portion of the runtime that corresponds to changes of state in the nodes of the flow ...; [0071])

13. **As to claim 11** (Original) (incorporating the rejection in claim 1), O'Farrel discloses the business process service debugger wherein said debugging means comprises means for detecting a location where the instance state is being stored (e.g., [0045] - ... in a portion of the runtime that corresponds to changes of state in the nodes of the flow ...; [0071])

14. **As to claim 40** (Previously Presented), O'Farrel discloses a method in a computer system for displaying on a display device a graphical debugging environment for a business process service (e.g., [0010] - ... a debugger allows debugging of applications using flow paradigm ...; [0011] - ... debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code ...) , the method comprising:

- obtaining design information about the business process service;
- displaying on the display device a graphical debugging environment showing the symbolic representation (e.g., [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Analyzing Decision Points in Business Processes*, Casati discloses:

- generating symbolic representation of the operation of the business process service according to the design information and tracking information (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 (i.e., the debugger) enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...); and
- obtaining tracking information about execution of the business process service (e.g., Fig. 5, elements 90 – Business Operational Intelligence Engine; 92-98 – databases; 100 – Business Process Data Warehouse; [0046], Lines 1-19 - ... a business operational intelligence engine 90 that supports the definition, execution, and tracking of business process ... by extracting the process execution data from databases 92-98 ...; Fig. 6; [0048] - An extract, transfer, and load (ETL) application 114 collects data

from the audit logs and loads the data into business process data warehouse 100 ...; [0050])

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Casati into the O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

The motivation is that it would further enhance the O'Farrel system by taking, advancing and/or incorporating Casati's system which offers significant advantages for a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision points in the business process as once suggested by Casati (e.g., [0010])

15. **As to claim 41** (Original) (incorporating the rejection in claim 40), O'Farrel discloses further comprising receiving runtime data for the business process service and presenting the runtime data on the display device (e.g., [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

16. **As to claim 42** (Previously Presented) (incorporating the rejection in claim 41), Casati discloses wherein the runtime data comprises historical message flow information including identification of run time messages that were constructed as a result of processing received messages, and further comprises order

information pertaining to the order in which different run time components were executed as a result of processing received messages (e.g., Fig. 5, elements 60 – the component of business platform; 70 – Web Server (i.e., a remote computer implementing the business process service); 90 – Business Operational Intelligence Engine; 100 – Business Process Data Warehouse; [0046], Lines 8-19 - … Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 …)

17. **As to claim 43** (Previously Presented) (incorporating the rejection in claim 40), O'Farrel discloses wherein the graphical debugging environment enables a user to place a breakpoint in the symbolic representation of the operation of the business process service (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)

18. **As to claim 44** (Previously Presented) (incorporating the rejection in claim 40), O'Farrel discloses the symbolic representation comprising symbols, wherein the graphical debugging environment also displays information about the symbols (e.g., [0043] - … further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

19. **As to claim 45** (Previously Presented) (incorporating the rejection in claim 40), O'Farrel discloses further comprising receiving input from an input device to place a break point proximate a symbol, and presenting a symbol representing the break point on the symbolic representation (e.g., [0012] - ... receiving user input to allow setting of at least one breakpoint in the application using the flow diagram; [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

20. **As to claim 60** (Previously Presented), O'Farrel discloses a computer-readable storage medium having computer-executable instructions for performing a method for displaying on a display device a graphical debugging environment for a business process service (e.g., [0010] - ... a debugger allows debugging of applications using flow paradigm ...; [0011] - ... debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code ...), the method comprising:

- obtaining design information about the business process service;
- obtaining configuration information about the business process service (e.g., [0033] - ... converts the graphically defined flow, into a plurality tokens, each token representative of a node on the graph. The tokens, in turn, may be stored as entries of a database ...; [0034] - Upon execution,

flow engine software such as that forming part of run-time code 52 may access the database to extract and interpret the tokens ...); and

- displaying on the display device a graphical debugging environment showing the symbolic representation (e.g., [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Analyzing Decision Points in Business Processes*, Casati discloses:

- generating symbolic representation of the operation of the business process service according to the design information and configuration information (e.g., [0010] - ... a graphical user interface configured to present to a user ... at different stages of the business process ...; [0025] ... any given service may be defined by a directed graph of business processes ...; Fig. 9; [0054] – [0056])

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Casati into the O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

The motivation is that it would further enhance the O'Farrel system by taking, advancing and/or incorporating Casati's system which offers significant

advantages for a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision points in the business process as once suggested by Casati (e.g., [0010])

21. **As to claim 61** (Previously Presented) (incorporating the rejection in claim 60), please refer to claim **41** as set forth accordingly.

22. **As to claim 62**, refer to above **claims 42**, accordingly.

23. **As to claims 63-65**, refer to above **claims 43-45**, accordingly.

24. Claims 12-20, 23-35, 37-39, and 46-59 are rejected under 35 U.S.C. 103(a) as being unpatentable over O'Farrel in view of Casati and further in view of Moser et al. (Pat. No. US 7,093,237 B2) (hereinafter 'Moser' - art made of record)

25. **As to claim 12** (Currently amended), O'Farrel discloses a system for remotely debugging a distributed transactional application, comprising:

- a server computer configured to execute an instance of a business process service, thereby generating runtime data (e.g., [0011] - ... debugging a software application comprising a plurality of processes

that asynchronously co-operate and whose execution is controlled through interaction with run-time code ...)

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Analyzing Decision Points in Business Processes*, Casati discloses:

- a client computer configured to execute a debugging user interface (UI) process that indirectly communicates with the server, establishes a communications connection with the server requests historical runtime data for at least one of the plurality of business processes (e.g., Fig. 5, elements 60 – the component of business platform; 70 – Web Server (i.e., indirectly communicating with the server); 90 – Business Operational Intelligence Engine; 100 – Business Process Data Warehouse; [0046], Lines 8-19 - ... Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 ...), and generates, based on the historical runtime data, a symbolic representation of the business service process showing any

debugging break points specified by a user (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 (i.e., the debugger) enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...);

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Casati into the O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

`The motivation is that it would further enhance the O'Farrel system by taking, advancing and/or incorporating Casati's system which offers significant advantages for a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision points in the business process as once suggested by Casati (e.g., [0010])

Furthermore, Casati discloses that a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision

points in the business process (e.g., [0010]) but O'Farrel and Casati do not explicitly disclose other limitations stated below.

However, in an analogous art of *High Performance Debugging in a Message Flow Environment*, Moser discloses an interceptor for monitoring the runtime data and, when a specified break point is identified, causing the server to enter or leave a debugging state (e.g., Col. 2, Lines 27-44 - ... providing a debug information to a developer in a message flow environment ... to define specified nodes to be breakpoint for the defined message flow ... if the node is defined to be a breakpoint, and if the node is defined a breakpoint, providing for the node to communicate debug information to the debug controller for display to the developer ...; Fig. 1, elements 20 – Debug Plugin Node; 22 – Debug Plugin Breakpoint Node; 16 – Debug Controller; 19 – GUI; Col. 4, Lines 1-12 – ... Fig. 1 shows debug controller 16 having a graphical user interface GUI 18. The developer user GUI 18 to pass events to debug controller (i.e., an interceptor) 16, which are used to define and manage a debug session in the defined message flow represented by nodes 10, 12, 14 ... each of debug nodes 20, 23 ... each of debug nodes 20, 22 would communicate directly with debug controller 16 when a message reached the debug node; Lines 13-27 - ... debug plugin node 20 is not defined to be a breakpoint whereas debug plugin node 22 is defined as a breakpoint ...)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Moser into the

O'Farrel-Casati's system to further provide other limitations stated above in the O'Farrel-Casati system.

The motivation is that it would further enhance the O'Farrel-Casati system by taking, advancing and/or incorporating Moser's system which offers significant advantages for a debugging facility for a message flow environment which permits high volumes of messages to be processed while debugging facilities are in place in a message flow, without significant detrimental overhead cost as once suggested by Moser (e.g., Col. 2, Lines 19-23)

26. **As to claim 13** (Original) (incorporating the rejection in claim 12), Casati discloses the system further comprising for storing business process service state information (e.g., [0046], Lines 8-19 - ... Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 ...)

Further, Casati discloses a database for receiving the runtime data (e.g., [0046], Lines 8-19 - ... Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received

by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 ...)

27. **As to claim 14** (Previously Presented) (incorporating the rejection in claim 13), O'Farrel discloses the system further comprising a display device, for displaying the symbolic representation (e.g., [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17) and a user input device, wherein the input device is used to specify debugging break points (e.g., [0012] - ... receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)

28. **As to claim 15** (Previously Presented) (incorporating the rejection in claim 14), O'Farrel discloses the system wherein the symbolic representation comprises presents a workflow representative of the program flow of the business process service (e.g., [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

29. **As to claim 16** (Previously Presented) (incorporating the rejection in claim 14), O'Farrel discloses the system wherein the display device further displays data representative of a message flow of the business process service (e.g., [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

30. **As to claim 17** (Previously Presented) (incorporating the rejection in claim 14), Casati discloses the system wherein the symbolic representation is presented according to stored state information (e.g., [0046], Lines 8-19 - ... Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 ...)

31. **As to claim 18** (Previously Presented) (incorporating the rejection in claim 12), Casati discloses the system wherein a message box database is coupled between the server and client computer and is configured for communicating debugging requests from the client computer (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...)

32. **As to claim 19** (Previously Presented) (incorporating the rejection in claim 18), Casati discloses the system wherein the UI process comprises an application program interface for communicating with the message box database (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 (i.e., the debugger) enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...)

33. **As to claim 20** (Previously Presented) (incorporating the rejection in claim 18), Casati discloses the system further comprising a tracking database to receive business process service tracking information, wherein the UI process comprises a UI component for communicating with the tracking database (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 (i.e., the debugger) enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...)

34. **As to claim 23** (Original) (incorporating the rejection in claim 12), Moser discloses the system wherein the interceptor is a component of a computer language that provides stored state tracking information (e.g., Col. 2, Lines 27-44 - ... providing a debug information to a developer in a message flow environment ... to define specified nodes to be breakpoint for the defined message flow ... if the node is defined to be a breakpoint, and if the node is defined a breakpoint, providing for the node to communicate debug information to the debug controller for display to the developer ...; Fig. 1, elements 20 – Debug Plugin Node; 22 – Debug Plugin Breakpoint Node; 16 – Debug Controller; 19 – GUI; Col. 4, Lines 1-12 – ... Fig. 1 shows debug controller 16 having a graphical user interface GUI 18. The developer user GUI 18 to pass events to debug controller (i.e., an interceptor) 16, which are used to define and manage a debug session in the defined message flow represented by nodes 10, 12, 14 ... each of debug nodes 20, 23 ... each of debug nodes 20, 22 would communicate directly with debug controller 16 when a message reached the debug node; Lines 13-27 - ... debug plugin node 20 is not defined to be a breakpoint whereas debug plugin node 22 is defined as a breakpoint ...)

35. **As to claim 24** (Original) (incorporating the rejection in claim 12), O'Farrel discloses the system wherein the UI process detects a location where the instance is being processed (e.g., [0045] - ... in a portion of the runtime that corresponds to changes of state in the nodes of the flow ...; [0071])

36. **As to claim 25** (Original) (incorporating the rejection in claim 12), O'Farrel discloses the system wherein the UI process detects a location where the instance state is being stored (e.g., [0045] - … in a portion of the runtime that corresponds to changes of state in the nodes of the flow …; [0071])

37. **As to claim 26** (Previously Presented), O'Farrel discloses a method for debugging an instance of a business process service running on a remote computer (e.g., [0010] - … a debugger allows debugging of applications using flow paradigm …; [0011] - … debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code …), comprising:

- receiving a debugging command generated by a user interacting with the GUI (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram);
- a direct client connection channel with the remote compute (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …);
- receiving a runtime request, generated by a user interacting with the GUI, for event information generated by execution of the instance of the business process service (e.g., [0012] - … receiving user input to allow setting of at least one breakpoint in the application using the flow diagram);

- sending the runtime request to the remote computer for processing by the remote computer (e.g., Fig. 9; [0083] - ... flow debugger 64 ... to attach to a known one of computing devices ... executing run-time code ... The computing device may be local to flow debugger 64 or remote, identified by a network address ...)

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Analyzing Decision Points in Business Processes*, Casati discloses:

- generating for display, in a graphical user interface (GUI), a symbolic representation of the business process service based on a correlation of information about the design and execution of the business process service (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Casati into the

O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

The motivation is that it would further enhance the O'Farrel system by taking, advancing and/or incorporating Casati's system which offers significant advantages for a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision points in the business process as once suggested by Casati (e.g., [0010])

Furthermore, Casati discloses that a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision points in the business process (e.g., [0010]) but O'Farrel and Casati do not explicitly disclose other limitations stated below.

However, in an analogous art of *High Performance Debugging in a Message Flow Environment*, Moser discloses causing an interceptor to monitor runtime data generated by the instance of the business process in accordance with the debugging command (e.g., Col. 2, Lines 27-44 - ... providing a debug information to a developer in a message flow environment ... to define specified nodes to be breakpoint for the defined message flow ... if the node is defined to be a breakpoint, and if the node is defined a breakpoint, providing for the node to communicate debug information to the debug controller for display to the developer ...; Fig. 1, elements 20 – Debug Plugin Node; 22 – Debug Plugin Breakpoint Node; 16 – Debug Controller; 19 – GUI; Col. 4, Lines 1-12 – ... Fig. 1

shows debug controller 16 having a graphical user interface GUI 18. The developer user GUI 18 to pass events to debug controller (i.e., an interceptor) 16, which are used to define and manage a debug session in the defined message flow represented by nodes 10, 12, 14 ... each of debug nodes 20, 23 ... each of debug nodes 20, 22 would communicate directly with debug controller 16 when a message reached the debug node; Lines 13-27 - ... debug plugin node 20 is not defined to be a breakpoint whereas debug plugin node 22 is defined as a breakpoint ...)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Moser into the O'Farrel-Casati's system to further provide other limitations stated above in the O'Farrel-Casati system.

The motivation is that it would further enhance the O'Farrel-Casati system by taking, advancing and/or incorporating Moser's system which offers significant advantages for a debugging facility for a message flow environment which permits high volumes of messages to be processed while debugging facilities are in place in a message flow, without significant detrimental overhead cost as once suggested by Moser (e.g., Col. 2, Lines 19-23)

38. **As to claim 27** (Original) (incorporating the rejection in claim 26), Casati discloses further comprising: querying a database containing a status of the business process service; displaying a query result on a display device; receiving user input with respect to the query result (e.g., [0045] - ... a conversation audit

log containing information about the context and context changes … is stored; [0046], Lines 23-37 - … business operational intelligence engine 90 (i.e., the debugger) enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies … in process executions and identify solutions … as perceived by external users …)

Further, O'Farrel discloses establishing the direct client connection channel in response to the user input (e.g., Fig. 9; [0083] - … flow debugger 64 … to attach to a known one of computing devices … executing run-time code … The computing device may be local to flow debugger 64 or remote, identified by a network address …)

39. **As to claim 28** (Original) (incorporating the rejection in claim 27), Casati discloses the information contained in the database is instance runtime data (e.g., [0046], Lines 8-19 - … Business operational intelligence engine 90 has the ability to log information about the business processes it supports, including, for example, the start and completion time of each activity, its input and output data, the resource that executed it, as well as every event (message) sent or received by a process. The audit log data maintained by the components of business process platform 60 may be stored in respective databases 92, 94, 96, 98 …)

40. **As to claim 29** (Original) (incorporating the rejection in claim 27), Casati discloses the information contained in the database is instance tracking data

(e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 (i.e., the debugger) enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...)

41. **As to claim 30** (Previously Presented) (incorporating the rejection in claim 26), O'Farrel discloses further comprising: creating the business process service using a process designer (e.g., [0032] - ... IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

Further, Casati discloses saving a business process service configuration and symbolic data in a database as information about the design of the business process service (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...); Furthermore, O'Farrel discloses displaying the symbolic representation on a display device according to the saved business

process service configuration and symbolic data; generating a runtime request based on the symbolic representation; and displaying a result of the runtime request on the display device (e.g., [0043] - ... further receives debug data for presentation by way of a graphical user interface (GUI) which is detailed with reference to Figs. 9-17)

42. **As to claim 31** (Previously Presented) (incorporating the rejection in claim 30), O'Farrel discloses the symbolic representation comprises a shape corresponding to an operation in the business process service (e.g., [0032] - ... IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

43. **As to claim 32** (Previously Presented) (incorporating the rejection in claim 30), O'Farrel discloses the symbolic representation comprises a workflow representation of the business process service (e.g., [0032] - ... IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

44. **As to claim 33** (Original) (incorporating the rejection in claim 30, O'Farrel discloses the saving step takes place in connection with compiling and deploying the business process service (e.g., [0032] - ... IBM WebSphere™ Studio

Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

45. **As to claim 34** (Original) (incorporating the rejection in claim 30), O'Farrel discloses the business process service is implemented in a computer language that provides stored state information (e.g., [0032] - ... IBM WebSphere™ Studio Application Developer Integration Edition (ESAD-IE) engine allows creation of applications in accordance with the flow paradigm, using a graphical user interface ...)

46. **As to claim 35** (Previously Presented) (incorporating the rejection in claim 26), O'Farrel discloses the debugging command is a break point (e.g., [0012] - ... receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)

47. **As to claim 37** (Previously Presented) (incorporating the rejection in claim 26), Moser discloses the runtime data is state information (e.g., Col. 3, Lines 1-6 - ... receiving user input to allow setting of at least one breakpoint in the application using the flow diagram)1-6 - ... communicating the runtime stack state to the debug controller)

48. **As to claim 38** (Original) (incorporating the rejection in claim 26), O'Farrel discloses further comprising detecting a location where the instance is being processed (e.g., [0045] - ... in a portion of the runtime that corresponds to changes of state in the nodes of the flow ...; [0071])

49. **As to claim 39** (Original) (incorporating the rejection in claim 26), O'Farrel discloses further comprising detecting a location where an instance state is being stored (e.g., [0045] - ... in a portion of the runtime that corresponds to changes of state in the nodes of the flow ...; [0071])

50. **As to claim 46** (Previously Presented), O'Farrel discloses a computer-readable storage medium having computer-executable instructions for performing a method for debugging an instance of a business process service running on a remote computer (e.g., [0010] - ... a debugger allows debugging of applications using flow paradigm ...; [0011] - ... debugging a software application comprising a plurality of processes that asynchronously co-operate and whose execution is controlled through interaction with run-time code ...), comprising:

- receiving a debugging command generated by a user interacting with the GUI (e.g., [0012] - ... receiving user input to allow setting of at least one breakpoint in the application using the flow diagram);
- establishing a direct client connection channel with the remote computer (e.g., Fig. 9; [0083] - ... flow debugger 64 ... to attach to a known one of

computing devices ... executing run-time code ... The computing device may be local to flow debugger 64 or remote, identified by a network address ...);

- receiving a runtime request; and sending the runtime request to the remote computer for processing by the remote computer (e.g., Fig. 9; [0083] - ... flow debugger 64 ... to attach to a known one of computing devices ... executing run-time code ... The computing device may be local to flow debugger 64 or remote, identified by a network address ...)

Further, O'Farrel discloses that a debugger allows debugging of applications using a flow paradigm (e.g., Abstract) but does not explicitly disclose other limitations stated below.

However, in an analogous art of *Analyzing Decision Points in Business Processes*, Casati discloses:

- generating for display, in a graphical user interface (GUI), a symbolic representation of the business process service based on a correlation of information about the design and execution of the business process services (e.g., [0045] - ... a conversation audit log containing information about the context and context changes ... is stored; [0046], Lines 23-37 - ... business operational intelligence engine 90 enables users to visualize a business process with a focus on a user-selected process entity .... enables users to analyze the information stored in business process data warehouse 100 to reveal problems and inefficiencies ... in process executions and identify solutions ... as perceived by external users ...)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Casati into the O'Farrel's system to further provide other limitations stated above in the O'Farrel system.

The motivation is that it would further enhance the O'Farrel system by taking, advancing and/or incorporating Casati's system which offers significant advantages for a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision points in the business process as once suggested by Casati (e.g., [0010])

Furthermore, Casati discloses that a graphical user interface configured to present to a user one or more rules correlating context data at different stages of the business process with business process outcomes at one or more decision points in the business process (e.g., [0010]) but O'Farrel and Casati do not explicitly disclose other limitations stated below.

However, in an analogous art of *High Performance Debugging in a Message Flow Environment*, Moser discloses causing an interceptor to monitor runtime data generated by the instance of the business process service in accordance with the debugging command (e.g., Col. 2, Lines 27-44 - ... providing a debug information to a developer in a message flow environment ... to define specified nodes to be breakpoint for the defined message flow ... if the node is defined to be a breakpoint, and if the node is defined a breakpoint, providing for the node to communicate debug information to the debug controller for display to

the developer ...; Fig. 1, elements 20 – Debug Plugin Node; 22 – Debug Plugin Breakpoint Node; 16 – Debug Controller; 19 – GUI; Col. 4, Lines 1-12 – ... Fig. 1 shows debug controller 16 having a graphical user interface GUI 18. The developer user GUI 18 to pass events to debug controller (i.e., an interceptor) 16, which are used to define and manage a debug session in the defined message flow represented by nodes 10, 12, 14 ... each of debug nodes 20, 23 ... each of debug nodes 20, 22 would communicate directly with debug controller 16 when a message reached the debug node; Lines 13-27 - ... debug plugin node 20 is not defined to be a breakpoint whereas debug plugin node 22 is defined as a breakpoint ...)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Moser into the O'Farrel-Casati's system to further provide other limitations stated above in the O'Farrel-Casati system.

The motivation is that it would further enhance the O'Farrel-Casati system by taking, advancing and/or incorporating Moser's system which offers significant advantages for a debugging facility for a message flow environment which permits high volumes of messages to be processed while debugging facilities are in place in a message flow, without significant detrimental overhead cost as once suggested by Moser (e.g., Col. 2, Lines 19-23)

51. **As to claims 47-55**, refer to above **claims 27-35**, accordingly.

52. **As to claim 56** (Previously Presented) (incorporating the rejection in claim 30), O'Farrel discloses the computer-readable storage medium, wherein the debugging command is a request for data regarding an instance of the business process service (e.g., [0043] - ... provides debug commands to the debug manger ...)

53. **As to claims 57-59**, refer to above **claims 37-39**, accordingly.

Response to Arguments

54. Applicant's arguments filed on April 14, 2009 for direct channel communication in the claims have been fully considered but they are not persuasive.

In the remarks, Applicant argues that, for examples:

(A.1) *O'Farrel* fails to teach or suggest the claimed direct channel communications (recited third full paragraph on page 14 in the REMARKS - emphasis added)

Examiner's response:

(R.1) *O'Farrel* clearly teaches, for example, "means for establishing a communication connection with a remote computer" (e.g., Fig. 9; [0083] – As illustrated in Fig. 9, flow debugger 64 initially presents a dialog 90 to attach to a known one of computing devices 12a, 12b, 12c, and 12d executing run-time code

52 … In response to selecting a computing device 12, flow debugger 64 queries this computing device 12 using a protocol address directing communications to communication layer 62 thus providing the request to debug manager 60 of the computing device 12 of interest ...) which meet the addressed limitations in claim 1.

Conclusion

55. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/
Ben C. Wang
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192